

Suggested reading

The following are synopses of four recent books on Java that may be of interest to readers of the *IBM Systems Journal*. Inquiries should be directed to the publishers cited.

Client/Server Programming with Java and CORBA, Second Edition, Robert Orfali and Dan Harkey, John Wiley & Sons, Inc., New York, 1998. 832 pp. (ISBN 0-471-24578-X). First and foremost, this book is about distributed objects written in Java**, with a strong influence from the CORBA** (Common Object Request Broker Architecture) distributed objects model. CORBA 2.0 is overviewed from a Java point of view, and it is compared and contrasted with other technologies that enable distributed objects in Java, such as sockets, CGI (Common Gateway Interface), servlets, RMI (remote method invocation), and DCOM** (Distributed Component Object Model). The book also includes an extensive tutorial on JDBC** (Java Database Connectivity). Toward the end of the book, the authors present a spectrum of Java components, from JavaBeans**, via CORBA components and Object Transaction Monitors, to Enterprise JavaBeans**. Thus, this book is targeted at Java programmers who need to understand client/server computing, distributed objects, CORBA/IIOP (CORBA/Internet Inter-Orb Protocol), JavaBeans and JDBC, and who also want to explore distributed components and Enterprise JavaBeans. This book is also for CORBA programmers who need to understand Java, JavaBeans, and Enterprise JavaBeans. Over half of this second edition contains entirely new material, including chapters on JavaBeans, CORBA beans, Enterprise JavaBeans, servlets, Java-to-IDL (interface definition language), IDL-to-Java, the CORBA Naming Service, and CORBA 3.0 features such as POA (portable object adaptor). In the Java landscape, this book provides a good charting of the Java for the Enterprise territory.

Java in a Nutshell: A Desktop Quick Reference, Second Edition, David Flanagan, O'Reilly & Associates, Sebastopol, CA, 1997. 610 pp. (ISBN 1-56592-262-X). The second edition of this book is the JDK** 1.1 (Java Development Kit 1.1) counterpart of its first edition. Its first three parts present an introduction to the language from the point of view of a C or C++ programmer. Given that the language has grown so much since the 1.0 release (not to mention the even larger growth it experiences in the 1.2 release), the examples that illustrate the introductory parts are not as prolific as they were in the first edition. Rather, the author uses a separate book to present a comprehensive set of Java 1.1 examples. Part five is the more substantial one of the book, presenting a reference guide to the classes in the Java 1.1 API (application programming interface) arranged by package. Each

package is introduced with a hierarchy of the classes and interfaces it contains. Entries for each class include a brief description of the class and its members, a source-code-like synopsis of its definition, and a number of optional cross-reference annotations that indicate other, related classes that may be of interest. Finally, a handy class, method, and field index is included that allows the reader to find where a given class or class member is defined. This book is a must-have reference for Java programmers. It is a very handy guide to the Java classes and interfaces.

The Java Developers Almanac, Patrick Chan, Addison-Wesley Publishing Co., Java Series, Reading, MA, 1998. 962 pp. (ISBN 0-201-37967-8). If what you need is a quick reference guide to the Java 1.2 (beta 3) core packages and some of the extension packages, this book is for you. In addition to listing packages, the *Almanac* contains a listing of all the classes in the packages it includes, arranged alphabetically by class name, as well as a section on miscellaneous topics, and an extensive cross-reference section. Given that Java is a living and evolving language, this book is intended to be updated periodically to reflect the newest version of the language and the contents of the core and extension packages. The author uses a quick reference format to describe each of the packages and classes listed in the book. This format encodes 13 pieces of descriptive information about a package or class at hand in a very concise and easy-to-read fashion. Miscellaneous topics include differences between 1.1 and 1.2, javadoc tags (documentation comment tags), applet tag parameters, system properties, language operators, etc.; PersonalJava** and JNI (Java Native Interface) APIs; and JVM (Java Virtual Machine) class file formats and bytecodes. Among the extension packages not included in the *Almanac* are EJB (Enterprise JavaBeans) and JDBC.

Although this book cannot be used as a vehicle for learning to use the Java packages and classes, it is a useful source of referential and cross-referential information on the Java APIs.

Concurrent Programming in Java: Design Principles and Patterns, Doug Lea, Addison-Wesley Publishing Co., Java Series, Reading, MA, 1997. 339 pp. (ISBN 0-201-69581-2). As Robin Milner has stated, "Communication and concurrency are complementary notions, both essential in understanding complex dynamic systems." If Java's support for concurrent programming is to be mastered, then an understanding of these notions is necessary prior to diving into the nuts and bolts of Java's constructs

©Copyright 1998 by International Business Machines Corporation.

used to initiate, control, and coordinate concurrent activities. It is tempting to jump right into writing concurrent programs without necessarily taking the time to understand the big picture of concurrent programming. This book is a valuable tool in resisting this temptation. It presents concrete examples of Java concurrent programs, after introducing the Java concurrency support constructs, while at the same time using these examples to address some of the harder issues underlying the notions of communication versus concurrency that are independent of Java. As an added bonus, each chapter that deals with a specific concurrency programming issue is annotated with references to further, more in-depth treatments of the issues at hand. In this book, the author uses a pattern-oriented approach to introduce and illustrate design techniques that address concurrent programming issues such as safety, liveness, state-dependent action, concurrency control, services in threads, flow, and coordinated interaction. As part of the Java series, this book should prove to be a faithful treatment of concurrency programming in Java and an indispensable resource for anyone programming in Java in general.

Ignacio Silva-Lepe
IBM T. J. Watson Research Center
Hawthorne
New York

**Trademark or registered trademark of Sun Microsystems, Inc., Object Management Group, or Microsoft Corporation.